

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-230393

(43) 公開日 平成7年(1995)8月29日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 11/28	A	7313-5B		
9/06	5 4 0 S	7629-5B		

審査請求 未請求 請求項の数 4 O L (全 16 頁)

(21) 出願番号 特願平6-20166

(22) 出願日 平成6年(1994)2月17日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 木村 浩一

東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

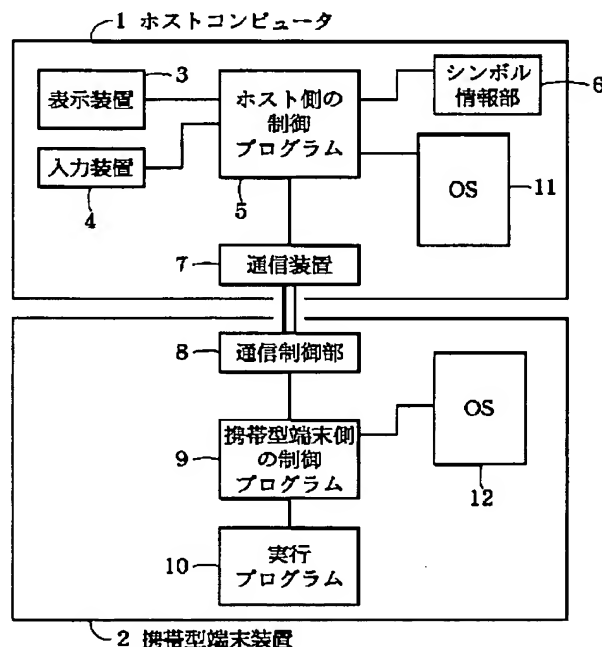
(74) 代理人 弁理士 小林 将高

(54) 【発明の名称】 情報処理システムおよび情報処理システムのプログラムデバッグ方法

(57) 【要約】

【目的】 実行プログラムのデバッグ処理時に可読性の高いソースプログラムのシンボル名でアドレス指定を行うことができる。

【構成】 入力装置4からの入力情報が前記シンボル名かどうかを判別する判別手段がシンボル名と判別した場合に、シンボル情報部6に記憶されたシンボル名を参照して実アドレス情報に変換する変換手段とをホストコンピュータ1に設け、入力装置4からの入力情報が前記シンボル名であると判別した場合に、制御プログラム5がシンボル情報部6に記憶されたシンボル名を参照して実アドレス情報に変換する構成を特徴とする。



【特許請求の範囲】

【請求項1】 一方の情報処理装置と他方の情報処理装置とが通信制御プログラムに基づいて相互に通信可能な情報処理システムにおいて、デバッグ対象の実行プログラムを実行する一方の情報処理装置の記憶領域の各アドレスに一致させるための所定のシンボル名を記憶するシンボル情報記憶手段と、一方の情報処理装置から他方の情報処理装置に転送される一方の情報処理装置で動作する実行プログラムを記憶する記憶手段を他方の情報処理装置側に設け、前記実行プログラムに記述されているシンボル名またはアドレス指定情報を入力する入力手段と、転送する情報を表示する表示手段とを双方の情報処理装置側に設け、前記入力手段からの入力情報が前記シンボル名かどうかを判別する判別手段と、この判別手段がシンボル名である場合に、前記シンボル情報記憶手段に記憶されたシンボル名を参照して実アドレス情報に変換する変換手段とを一方の情報処理装置に設けたことを特徴とする情報処理システム。

【請求項2】 デバッグ対象の実行プログラムを実行する一方の情報処理装置の記憶領域の各アドレスに一致させるための所定のシンボル名を記憶するシンボル情報記憶手段を有し、一方の情報処理装置と他方の情報処理装置とが通信制御プログラムに基づいて相互に通信可能な情報処理システムのプログラムデバッグ方法において、前記実行プログラムに記述されているシンボル名またはアドレス指定情報を入力する入力工程と、入力情報が前記シンボル名かどうかを判別する判別工程と、入力情報がシンボル名である場合に、前記シンボル情報記憶手段に記憶されたシンボル名を参照して実アドレス情報に変換する変換工程とを有し、一方または他方の情報処理装置側から入力されるシンボル名に応じて一方の情報処理装置で実行される実行プログラムの実アドレスを生成することを特徴とする情報処理システムのプログラムデバッグ方法。

【請求項3】 デバッグ対象の実行プログラムを機械語レベルのプログラムで記述していることを特徴とする請求項2記載の情報処理システムのプログラムデバッグ方法。

【請求項4】 デバッグ対象の実行プログラムを中間言語により翻訳・実行するレベルのプログラムで記述していることを特徴とする請求項2記載の情報処理システムのプログラムデバッグ方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、各情報処理装置が相互に通信可能な情報処理システムに係り、入力されるシンボル情報に従って生成されるデバッグ対象の実行プログラムの実アドレスに従ってプログラムのデバッグを実行する情報処理システムおよび情報処理システムのプログラムデバッグ方法に関するものである。

【0002】

【従来の技術】従来、プログラム開発環境下において、デバッガと呼ばれるプログラムと開発対象となるプログラムとが共に実行のプラットフォームである情報処理装置、例えば携帯型端末装置上に置かれていた。

【0003】

【発明が解決しようとする課題】また、デバッガに対する入力を外部の機器から行う通信によって外部機器から行うということも行われていたが、シンボルの情報が削除された状態で作業を行うため、実行するプログラムをCPUが直接実行する機械語のレベルでしか扱えなかった。

【0004】別のアプローチとして、パーソナルコンピュータ等でプログラムの環境をシミュレートすることによりデバッグを行うという手段があるが、これはシミュレートするためのプログラムが別途必要であるということ、シミュレータではプログラムのロジック検証は行えるが、対象となる機械に付属する周辺機器の制御を含んだシステムのデバッグが行えないという欠点があった。

【0005】この様に、従来携帯型端末上で動作するプログラムを作成する途中でのデバッグ作業は、携帯型端末のメモリ容量等の理由により、シンボル名の情報を実行プログラムに付加することができなかったり、また、携帯型端末上で動作するデバッガを通常のコンピュータで使用されているような高機能のものにするのが難しいという理由で、デバッグの対象となるプログラムは、実行プログラムに変換される以前のソースプログラムに比べると可読性の低い、機械語に近いレベル状態のプログラムに対して行わなければならなかった。

【0006】従来の情報処理装置を利用した開発システムで行われるデバッグは、実行プログラムの中に特定の変数の値を表示する命令を挿入して実行時に表示させるスナップショットと呼ばれる手法を用いていた。また、機械語レベルのデバッガを使用したり、デバッグ対象機器がプログラム開発時と異なるキーボードレイアウトのため使いにくい、あるいはシンボル名の入力を行うことができない等の問題点があった。

【0007】本発明は、上記の問題点を解消するためになされたもので、本発明の第1の目的は、一方の情報処理装置で動作する実行プログラムを他方の情報処理装置に転送し、いずれかの情報処理装置より入力される情報がソースプログラム中の各シンボル名である場合に、記憶されたシンボル情報を参照して一方の情報処理装置のデバッグ対象のプログラムが実行される記憶領域の実アドレスに変換することにより、デバッグ対象の一方の情報処理装置における実行プログラムのデバッグ処理時に可読性の高いソースプログラムのシンボル名でアドレス指定を行える情報処理システムを提供することである。

【0008】本発明の第2の目的は、一方の情報処理装置で動作する実行プログラムを他方の情報処理装置に転

3

送し、いずれかの情報処理装置より入力される情報がソースプログラム中の各シンボル名である場合に、記憶されたシンボル情報を参照して一方の情報処理装置のデバッグ対象のプログラムが実行される記憶領域の実アドレスに変換して実行プログラムのデバッグを実行処理することにより、デバッグ対象の一方の情報処理装置における実行プログラムのデバッグ処理時に可読性の高いソースプログラムのシンボル名でアドレス指定しながら効率よく実行プログラムのデバッグを行える情報処理システムのプログラムデバッグ方法を提供することである。

【0009】本発明の第3の目的は、一方の情報処理装置で動作する実行プログラムを他方の情報処理装置に転送し、いずれかの情報処理装置より入力される情報がソースプログラム中の各シンボル名である場合に、記憶されたシンボル情報を参照して一方の情報処理装置のデバッグ対象のプログラムが実行される記憶領域の実アドレスに変換して機械語のレベルの実行プログラムのデバッグを実行処理することにより、デバッグ対象の一方の情報処理装置における実行プログラムのデバッグ処理時に可読性の高いソースプログラムのシンボル名でアドレス指定しながら効率よく可読性の低い実行プログラムのデバッグを行える情報処理システムのプログラムデバッグ方法を提供することである。

【0010】本発明の第4の目的は、一方の情報処理装置で動作する実行プログラムを他方の情報処理装置に転送し、いずれかの情報処理装置より入力される情報がソースプログラム中の各シンボル名である場合に、記憶されたシンボル情報を参照して一方の情報処理装置のデバッグ対象のプログラムが実行される記憶領域の実アドレスに変換して中間言語レベルの実行プログラムのデバッグを実行処理することにより、デバッグ対象の一方の情報処理装置における中間言語レベルの実行プログラムのデバッグ処理時に可読性の高いソースプログラムのシンボル名でアドレス指定しながら効率よく実行プログラムのデバッグを行える情報処理システムのプログラムデバッグ方法を提供することである。

【0011】

【課題を解決するための手段】第1の発明は、一方の情報処理装置と他方の情報処理装置とが通信制御プログラムに基づいて相互に通信可能な情報処理システムにおいて、デバッグ対象の実行プログラムを実行する一方の情報処理装置の記憶領域の各アドレスに一致させるための所定のシンボル名を記憶するシンボル情報記憶手段と、一方の情報処理装置から他方の情報処理装置に転送される一方の情報処理装置で動作する実行プログラムを記憶する記憶手段を他方の情報処理装置側に設け、前記実行プログラムに記述されているシンボル名またはアドレス指定情報を入力する入力手段と、転送する情報を表示する表示手段とを双方の情報処理装置側に設け、前記入力手段からの入力情報が前記シンボル名かどうかを判別す

4

る判別手段と、この判別手段がシンボル名である場合に、前記シンボル情報記憶手段に記憶されたシンボル名を参照して実アドレス情報に変換する変換手段とを一方の情報処理装置に設けたものである。

【0012】第2の発明は、デバッグ対象の実行プログラムを実行する一方の情報処理装置の記憶領域の各アドレスに一致させるための所定のシンボル名を記憶するシンボル情報記憶手段を有し、一方の情報処理装置と他方の情報処理装置とが通信制御プログラムに基づいて相互に通信可能な情報処理システムのプログラムデバッグ方法において、前記実行プログラムに記述されているシンボル名またはアドレス指定情報を入力する入力工程と、入力情報が前記シンボル名かどうかを判別する判別工程と、入力情報がシンボル名である場合に、前記シンボル情報記憶手段に記憶されたシンボル名を参照して実アドレス情報に変換する変換工程とを有し、一方または他方の情報処理装置側から入力されるシンボル名に応じて一方の情報処理装置で実行される実行プログラムの実アドレスを生成するものである。

【0013】第3の発明は、デバッグ対象の実行プログラムを機械語レベルのプログラムで記述するように構成したものである。

【0014】第4の発明は、デバッグ対象の実行プログラムを中間言語により翻訳・実行するレベルのプログラムで記述するように構成したものである。

【0015】

【作用】第1の発明においては、入力手段からの入力情報が前記シンボル名であると判別手段が判別した場合に、変換手段が前記シンボル情報記憶手段に記憶されたシンボル名を参照して実アドレス情報に変換して、デバッグ対象の一方の情報処理装置における実行プログラムのデバッグ処理時に可読性の高いソースプログラムのシンボル名でアドレス指定を行うものである。

【0016】第2の発明においては、実行プログラムに記述されているシンボル名またはアドレス指定情報が一方または他方の情報処理装置から入力されたと判別した場合に、前記シンボル情報記憶手段に記憶されたシンボル名を参照して実アドレス情報に変換し、一方または他方の情報処理装置側から入力されるシンボル名に応じて一方の情報処理装置で実行される実行プログラムの実アドレスを生成して、デバッグ対象の一方の情報処理装置における実行プログラムのデバッグ処理時に可読性の高いソースプログラムのシンボル名でアドレス指定しながら効率よく実行プログラムのデバッグを行うものである。

【0017】第3の発明においては、一方の情報処理装置のデバッグ対象のプログラムが入力情報を実行される記憶領域の実アドレスに変換して機械語のレベルの実行プログラムのデバッグを実行処理することにより、デバッグ対象の一方の情報処理装置における実行プログラム

5

のデバッグ処理時に可読性の高いソースプログラムのシンボル名でアドレス指定しながら効率よく可読性の低い実行プログラムのデバッグを行うものである。

【0018】第4の発明においては、記憶されたシンボル情報を参照して一方の情報処理装置のデバッグ対象のプログラムが入力情報を実行される記憶領域の実アドレスに変換して中間言語レベルの実行プログラムのデバッグを実行処理することにより、デバッグ対象の一方の情報処理装置における中間言語レベルの実行プログラムのデバッグ処理時に可読性の高いソースプログラムのシンボル名でアドレス指定しながら効率よく実行プログラムのデバッグを行うものである。

【0019】

【実施例】

【第1実施例】図1は本発明の第1実施例を示す情報処理システムの構成を示す外観斜視図である。

【0020】図において、1はホストコンピュータで、後述するコントロールユニットを有し、プログラムに基づいて種々のデータ処理を行う。2は携帯型端末装置で、ホストコンピュータ1とインタフェースケーブルI/Fを介して接続される。3は表示装置で、入力装置4から入力されたデータを表示するとともに、システム状態をプログラムに応じて表示する。なお、入力装置4は、キーボード4aとポインティングデバイス4bから構成されている場合を示す。

【0021】本実施例における第2の情報処理装置は、携帯可能なターミナル装置（携帯型の端末装置2）で構成しており、該第2の情報処理装置が後述するプログラムデバッグ方法を使用してプログラムデバッグ処理を行うものである。

【0022】また、本実施例における第1の情報処理装置として、ホストコンピュータ本体1を使用し、第2の情報処理装置として、携帯型端末装置（ターミナル装置）2を使用し、インタフェースケーブルI/Fを介してデータ通信する構成としている。

【0023】また、携帯可能なターミナル装置で構成された第2の情報処理装置が所定のインタフェースケーブルI/Fを介してプログラムデバッグ方法を使用してデバッグ処理を行うものである。本実施例では、所定のインタフェースケーブルが汎用I/FケーブルとしてのRS232C等で構成されているので、種々のホストコンピュータと携帯型端末装置2とを接続して、後述するプログラムデバッグ方法を使用したプログラムのデバッグ処理が可能となる。

【0024】さらに、携帯可能なターミナル装置で構成された第2の情報処理装置が所定のネットワークおよび所定の通信インタフェースケーブルを介してプログラムデバッグ方法を使用してデバッグ処理を行うものである。本実施例では、所定のネットワークが汎用LANが、イーサネット（商品名）、ネットウエア（商品

6

名）、ローカルトーク（商品名）等で構成され、ホストコンピュータと携帯型端末装置2とが遠隔されて使用される環境においても、後述するプログラムデバッグ方法を使用したプログラムのデバッグ処理が可能となる。

【0025】また、携帯可能なターミナル装置で構成された第2の情報処理装置が所定の通信プロトコルに基づいてプログラムデバッグ方法を使用してデバッグ処理を行うものである。本実施例では、リモコン通信に使用される無線通信原理を利用した通信インタフェースを備えて、ホストコンピュータと携帯型端末装置2とが直接ケーブル接続することなく、遠隔されて使用される環境においても、後述するプログラムデバッグ方法を使用したプログラムのデバッグ処理が可能となる。

【0026】図2は、図1に示したコンピュータシステムの制御構成を説明するブロック図であり、図1と同一のものには同一の符号を付してある。

【0027】図において、5はホスト側の制御プログラム（携帯型端末装置2との通信を行うプログラム）で、オペレーティングシステム（OS）11にその実行状態が管理されている。ホスト側の制御プログラム5は、入力装置4からのユーザの入力の読み取りと、表示装置3に対する出力と、携帯型端末側の制御プログラム9に対する通信動作を行う。

【0028】6はシンボル情報部で、実行形態のプログラム10に対応するシンボル情報を記憶している。12は前記携帯型端末装置2のOSで、携帯型端末側の制御プログラム9、実行プログラム10の実行を管理する。7、8は通信制御部で、通信経路（例えば所定のインタフェース（汎用としては、例えばRS232C）、LAN（ローカルエリアネットワーク）、無線等）を介してホストコンピュータ1で開発した実行プログラムを携帯型端末装置2に対する転送を制御する。携帯型端末側の制御プログラム9は、通信経路を通じてホスト側の制御プログラム5から入力されたコマンド、データに従って各種の動作を行い、また、デバッグの対象となる実行プログラムに対する監視の処理を行う。

【0029】この図に示されるように、開発のホストとなるホストコンピュータ1には携帯型端末装置2との通信を行うための制御プログラム5と、実行形態のプログラム10に対応するシンボル情報部6が置かれ、携帯型端末には開発ホストからダウンロードした実行形態のプログラム10と、ホストとの通信と実行プログラム10の監視および制御を行う制御プログラム9が置かれている。なお、実行プログラム10は、制御プログラム9の制御の下で動作し、ユーザが任意の操作を外部から加えることが可能である。

【0030】これにより、ホストコンピュータ1に入力されたコマンド、データは、ホスト側の制御プログラム5によって解析され、データ中にシンボル名があったならば、シンボル情報中で入力されたシンボル名と同一の

7

ものを検索（後述する検索処理ルーチンに従って実行される）し、見つかった場合にはそのシンボルに対応するアドレスを取得する。この時、ユーザの入力ミス等で検索が失敗した場合にはエラーとなる。

【0031】また、変換後のアドレスによって通信経路を通じてホスト側制御プログラム5に対しての指示を行う。

【0032】端末側の制御プログラム9は、受信したコマンドに対応する動作を行い、指定のコマンドが何等かの結果をホスト側の制御プログラム5に対して返さなければならない場合には、端末側の制御プログラム9はホスト側の制御プログラム5に対して送信を行う。ここで必要があれば、ホスト側制御プログラム5は受信したデータを加工して出力することにより、ユーザが認識しやすいようにすることも可能である。

【0033】以下、本発明の第1の発明に係る各手段と第1実施例との各部との対応およびその作用について説明する。

【0034】第1実施例において、一方の情報処理装置（ホストコンピュータ1）と他方の情報処理装置（携帯型端末装置2）とが通信制御プログラム（各制御プログラム5、9）に基づいて相互に通信可能な第1の発明に係る情報処理システムは、デバッグ対象の実行プログラムを実行する一方の情報処理装置の記憶領域の各アドレスに一致させるための所定のシンボル名を記憶するシンボル情報記憶手段（シンボル情報部6）と、一方の情報処理装置から他方の情報処理装置に転送される一方の情報処理装置で動作する実行プログラムを記憶する記憶手段（実行プログラム10）を他方の情報処理装置側に設け、前記実行プログラム10に記述されているシンボル名またはアドレス指定情報を入力する入力手段（入力装置4等）と、転送する情報を表示する表示手段（表示装置3）とを双方の情報処理装置側（図2に示した例では、ホストコンピュータ1側のみに設けられる構成となっている）に設け、例えば入力装置4からの入力情報が前記シンボル名かどうかを判別する判別手段（後述するCPU13が制御プログラム5に基づいて判別する手段）と、この判別手段がシンボル名である場合に、シンボル情報部6に記憶されたシンボル名を参照して実アドレス情報に変換する変換手段（後述するCPU13が制御プログラム5に基づいて変換する）とを一方の情報処理装置に設け、判別手段が入力装置4からの入力情報が前記シンボル名であると判別した場合に、制御プログラム5がシンボル情報部6に記憶されたシンボル名を参照して実アドレス情報に変換して、デバッグ対象の一方の情報処理装置における実行プログラムのデバッグ処理時に可読性の高いソースプログラムのシンボル名でアドレス指定を行うものである。

【0035】図3は、図1に示したコンピュータシステムの要部詳細ブロック図である。

8

【0036】図において、13はCPUで、RAM16に常駐するOSの管理下で実行されるプログラムに基づいて表示機器14、入力機器15、通信ポート17、外部記憶装置18等のアクセスを制御している。19は携帯型端末装置2側のCPUで、ROM23に記憶されたデータ処理プログラムをRAM22に常駐するOSの管理下で実行して、表示機器20にその処理結果を表示したり、入力機器24からの入力を処理する。また、通信ポート21とホストコンピュータ1側の通信ポート17とを介して通信を行い、携帯型端末装置2のプログラムのデバッグを行う。

【0037】なお、ホスト側の制御プログラム5は入力機器15から入力されたデータあるいは通信ポート17から入力されたデータを読み出し、必要に応じて表示機器14に表示する。また、実行プログラム10と携帯型端末装置2側の制御プログラム9は内部記憶装置としてのRAM22に展開される。

【0038】図4は、図2に示したシンボル情報部6に記憶されたシンボル情報の一例を示す図である。

【0039】この図に示されるように、本実施例のシンボル情報は、シンボル名が対応するアドレスが実記憶領域上において何番地であるかというアドレス情報をプログラムの特定部分を基準として表したデータ構造となっている。

【0040】このため、プログラムが実行される際には、プログラムが実記憶領域に置かれるアドレスは一意に決定できる状態となる。これにより、決定したアドレスの特定部分を基準として、上述したシンボル情報とを突き合わせることによって、あるシンボルが実際には携帯型端末のメモリ上で何番地に対応するかということを決定することができる。また、ユーザがシンボルを入力した際には、ホストコンピュータ側の制御プログラム5は、シンボル情報よりシンボルを検索し、そのシンボルが示すアドレスを取得する。

【0041】図5は、図2に示したホストコンピュータ1から転送されるコマンドパケットの一例を示す図である。

【0042】この図に示される様に、ホストコンピュータ1と携帯型端末装置2との通信時には、先頭にある1バイトのコマンド指定と、それに続くパラメータフィールドから構成されるコマンドパケットを作成して携帯型端末装置2側に転送される。

【0043】図6は、図2に示したホストコンピュータ1のコマンド処理手順の一例を示すフローチャートである。なお、(1)～(3)は各ステップを示す。

【0044】まず、ユーザが入力装置4からコマンドを入力すると(1)、該コマンドが終了コマンドかどうかを判定し(2)、YESならば処理を終了し、NOならば入力されたコマンドに対応した動作を行い(3)、ユーザが終了を示すコマンドが入力されるまで、コマンド

実行を繰り返す。これにより、実行プログラム10は、あらかじめホストコンピュータ1の実行形式へと変換され、携帯型端末装置2へ転送されて行く。

【0045】以下、図7に示すプログラム転送指定画面および図8に示すフローチャートを参照しながらデバッグ対象プログラム指定処理動作について説明する。

【0046】図8は本発明の一実施例を示す情報処理装置のプログラムデバッグ方法における第1の対象プログラム指定処理手順の一例を示すフローチャートである。なお、(1)～(4)は各ステップを示す。

【0047】ユーザがプログラムを指定すると、ホスト側の制御プログラム5は通信を行うためのデータパケットを図5に示したプロトコルに従って作成し、携帯型端末装置2側の制御プログラム9に対して送信する。

【0048】制御プログラム9は、指定されたプログラムを実行可能状態にするように、2次記憶装置としてのROM23から実メモリとしてのRAM22へ展開する等、操作を行う。

【0049】ここで、CPU19は指定されたプログラムを実行することができるかどうかを判定し(1)、ROM23に指定プログラムが存在しない場合または実行に必要なメモリ容量をRAM22に確保できない等の場合により即実行可能な状態に設定できなかったと判定した場合は、ステップ(4)に進み、その旨をエラーを表すコードとともに携帯型端末装置2からホストコンピュータ1へ送信して、処理を終了する。

【0050】一方、ステップ(1)の判定でYESの場合は、指定されたプログラムが展開された領域の先頭アドレスをホストコンピュータ1側の制御プログラム5に対して送信する(2)。次いで、ホストコンピュータ1側の制御プログラム5は、以後の作業の高速化を目的として、シンボル情報のプログラムの先頭をベースアドレスに構成されているアドレス部分を実アドレスに変換するシンボル情報の再構成を行い(3)、処理を終了する。

【0051】ただし、ユーザによるアドレスの指定がシンボルではなく、アドレスを表す数値によって指定された場合には、上記の変換動作は行われない。

【0052】以下、図9に示すフローチャートに示すプログラムデバッグ方法を参照しながら、第2、第3の発明の構成と第1実施例との対応およびその作用について説明する。

【0053】第2の発明によれば、実行プログラムに記述されているシンボル名またはアドレス指定情報を入力する入力工程と、入力情報が前記シンボル名かどうかを判別する判別工程と、入力情報がシンボル名である場合に、シンボル情報記憶手段に記憶されたシンボル名を参照して実アドレス情報に変換する変換工程とを実行して、前記実行プログラムに記述されているシンボル名またはアドレス指定情報が一方または他方の情報処理装置

から入力されたものと判別した場合に、前記シンボル情報記憶手段に記憶されたシンボル名を参照して実アドレス情報に変換し、一方または他方の情報処理装置側から入力されるシンボル名に応じて一方の情報処理装置で実行される実行プログラムの実アドレスを生成して、デバッグ対象の一方の情報処理装置における実行プログラムのデバッグ処理時に可読性の高いソースプログラムのシンボル名でアドレス指定しながら効率よく実行プログラムのデバッグを行うものであり、図2、図3との構成を備える第2の情報処理装置として機能する携帯型端末装置2からRS232C等のインタフェース(通信手段)を介して前記第1の情報処理装置として機能するホストコンピュータ1に入力された(図9に示すステップ(1))コマンドを、前記シンボル情報部6に記憶される前記参照情報(シンボル名に対応する変換アドレスが対となって記憶されている)情報に基づいて前記シンボル名によるアドレス指定を前記第1の情報処理装置の実アドレス(例えばシンボル名「AAA」であれば実アドレス「8000h」)に変換し(図9に示すステップ(3))ながら転送し、該コマンドに対して前記第1の情報処理装置から転送される(図9に示すステップ(4))デバッグ情報を第2の情報処理装置の表示部(例えば液晶ディスプレイパネル)に表示するので、実行プログラムのデバッグの際におけるアドレス指定等をソースプログラムのシンボル名で指定することが可能となる。

【0054】第3の発明によれば、一方の情報処理装置(ホストコンピュータ)のデバッグ対象のプログラムが入力情報(シンボル名等)を実行される記憶領域の実アドレスに変換して機械語のレベルの実行プログラムのデバッグを実行処理することにより、デバッグ対象の一方の情報処理装置における実行プログラムのデバッグ処理時に可読性の高いソースプログラムのシンボル名でアドレス指定しながら効率よく可読性の低い実行プログラムのデバッグを行うものである。

【0055】図9は本発明の一実施例を示す情報処理装置のプログラムデバッグ方法における第1のブレークポイント設定処理手順の一例を示すフローチャートである。なお、(1)～(4)は各ステップを示す。

【0056】ホストコンピュータ1に対しては、サブルーチンの名称等のシンボルを使用して、このシンボルに相当するアドレスに実行が移ったならば、そこで実行を中断するようにユーザが指令する(1)。これにより、ホストコンピュータ1側の制御プログラム5は、入力装置4から入力されたデータの読み取り動作を行い、シンボルによるアドレス指定がなされているかどうかを判定し(2)、NOならばステップ(4)以降に進み、YESならば制御プログラム5は、読み取ったデータを入力データ中のアドレス指定に使用されているシンボルをシンボル情報を参照して実番地に変換する(3)。変換した後、実アドレスによってプログラムのブレークポイ

ントを設定指示するように、プロトコルに従って作成したデータパケットを携帯型端末装置2の制御プログラム9に対して送信して(4)、処理を終了する。

【0057】これを受けて、携帯型端末装置2の制御プログラム9は受信したデータに従ってブレークポイントを設定する。以下、図10に示すブレークポイント確認処理を実行する。

【0058】図10は本発明の一実施例を示す情報処理装置のプログラムデバッグ方法における第1のブレークポイント確認処理手順の一例を示すフローチャートである。なお、(1)～(3)は各ステップを示す。

【0059】ホストコンピュータ1から転送されたパケットを受信したら、携帯型端末装置2側の制御プログラム9は、ブレークポイントが設定されているかどうかを判定し(1)、NOならば処理を終了し、YESならば携帯型端末装置2からホストコンピュータ1の制御プログラム5に対してアドレスを送信し(2)、受信したらブレークポイントのアドレスのデータをホストコンピュータ1側の表示装置14に表示して(3)、処理を終了する。

【0060】図11は本発明の一実施例を示す情報処理装置のプログラムデバッグ方法におけるメモリ参照処理手順の一例を示すフローチャートである。なお、(1)～(9)は各ステップを示す。

【0061】ホストコンピュータ1側の制御プログラム5が入力装置4からユーザにより入力されると(1)、入力されたアドレスまたはアドレスを示すシンボルを読み取り、それがシンボルで表現されているかどうかを判定し(2)、NOならばステップ(4)以降に進み、YESならば上述したように実アドレスへと変換する(3)。ただし、シンボルとして、CPUのレジスタ名を記述することもでき、その場合には、シンボル情報を用いての変換はされず、実アドレスへの変換は、携帯型端末装置2側の制御プログラム9によって実際のレジスタ値を反映して行われる。次いで、メモリ内容をホストコンピュータ1に対して送信するよう指示するコマンドと、その開始アドレス、必要があれば送信する範囲をプロトコルに従ってデータパケットを作成し、通信経路を通じて携帯型端末装置2に送信する(4)。データパケットを受信した携帯型端末装置2側の制御プログラム9は、データパケットから指定アドレスと範囲を読み出し(5)、指定範囲中かどうかを判定し(6)、NOならば処理を終了し、YESならば現在のアドレスポイントが指示するデータを読み出し(7)、アドレスポイントをインクリメントし(8)、読み出したデータを端末からホストコンピュータ1へ送信して(9)、ステップ(6)に戻る。

【0062】これにより、携帯型端末装置2側の制御プログラム9は、データパケット中のアドレス指定に従ってそのアドレスのデータをホストコンピュータ1側に送

信する範囲が指定されていれば、そのアドレスまで、指定されていない場合には、あらかじめ定められているデフォルトの数値までの範囲のデータをホスト側制御プログラム5に送信する。ホスト側の制御プログラム5は、受信したデータをホストコンピュータ1側の表示装置3に対して出力する。この時、ホストコンピュータ1側の制御プログラム5は、ユーザ選択によって、バイト単位、ワード単位、文字列とみなす(コントロールコード等の表示不可能な文字コードは適当な文字に置換する)等のフォーマットで出力することができるよう構成されている。

【0063】図12は本発明の一実施例を示す情報処理装置のプログラムデバッグ方法におけるメモリ書換え処理手順の一例を示すフローチャートである。なお、(1)～(12)は各ステップを示す。

【0064】ユーザが入力装置4を通じてアドレスもしくは範囲が入力されると(1)、該アドレスまたは該アドレスを表すシンボルを読み取り、シンボルによるアドレス指定がなされているかどうかを判定し(2)、NOならばステップ(4)以降に進み、YESならばシンボルの実アドレスへの変換を行い(3)、ホストコンピュータ1から携帯型端末装置2に対して開始アドレスを送信する(4)。携帯型端末装置2側の制御プログラム9は、指定されたアドレスの内容をホストコンピュータ1側に対して送信する(5)。該データを受信したら、ホストコンピュータ1側の表示装置3に該データを表示する(6)。次いで、書き換えデータを入力し(7)、入力されたデータが終了指示であるかどうかを判断し(8)、YESの場合には処理を終了し、そうでない場合は、現在のアドレスポイントのデータを変更し(9)、アドレスポイントをインクリメントし(10)、変更データを携帯型端末装置2からホストコンピュータ1へ送信し(11)、変更データをホストコンピュータ1の表示装置3に表示して(12)、ステップ(5)に戻る。

【0065】これにより、ユーザが変更の終了を示す入力を行わない限り、ユーザの入力データで携帯型の端末装置2の指定のアドレスの内容を書き換え、指定アドレスを変更するという動作を継続する。そして、新たなデータをメモリへ転送した後に、そのデータをホスト側に送り返すホスト側の制御プログラム5は、変更後のデータを受信した後に、変更前のデータと、変更後のデータをホスト側の表示装置3に出力する。

【0066】なお、入力されたデータがデータとして認められないものであった場合、ホストコンピュータ1側の制御プログラムが、その旨をユーザに通知し、この時携帯型端末装置2側の制御プログラム9に対して変更データの送信等は行わない。また、入力するデータをユーザの指定によってバイトデータ、ワードデータ、文字列データ等のフォーマットで入力することが可能に構成さ

れており、ホストコンピュータ1側の制御プログラム5は、各種のフォーマットで入力されたデータを統一的なフォーマットに変換し、変換した結果を携帯型端末装置2側に送信する。

【0067】図13は本発明の一実施例を示す情報処理装置のプログラムデバッグ方法におけるプログラム実行処理手順の一例を示すフローチャートである。なお、

(1)～(6)は各ステップを示す。

【0068】ユーザが実行開始番地の指定/無指定のいずれを指定しているかを判定し(1)、実行開始番地が指定されなかった場合は、開始番地をデフォルトアドレスとして(3)、ステップ(4)以降に進む。

【0069】一方、ステップ(1)の判定でYESの場合は指定番地を開始番地とし(2)、ホストコンピュータ1から携帯型端末装置2にコマンド、実行開始番地の指定の有無、実行開始番地をプロトコルに従ってデータパケットに作成し、携帯型端末装置2に送信する(4)。次いで、携帯型端末装置2側の制御プログラム9は受信したデータに従ってプログラムを実行させる(5)。そして、プログラム実行終了後または中断後の状態を携帯型端末装置2からホストコンピュータ1に送信して(6)、処理を終了する。なお、ステップ(1)の判定で、実行開始番地指定がなされなかった場合、例えばロード直後であれば、プログラムのエントリポイントが、ブレークポイントでブレークした後の状態であれば、前回ブレークした次の命令のある番地を開始番地として動作する。

【0070】図14は本発明の一実施例を示す情報処理装置のプログラムデバッグ方法におけるステップ実行処理手順の一例を示すフローチャートである。なお、(1)～(9)は各ステップを示す。

【0071】まず、ユーザから実行開始アドレスの指定がなされているかどうかを判断し(1)、指定がなされていない場合には、デフォルトアドレスを開始番地として(3)、ステップ(4)以降に進み、指定されていると判定した場合には、指定番地を開始番地とし(2)、ホストコンピュータ1から携帯型端末装置2にコマンド、実行開始番地をパケットとして送信する(4)。これに応じて携帯型端末装置2側の制御プログラム9は指定の番地にある1命令を実行し(5)、デフォルトの開始アドレスを命令実行後のアドレスにして(6)、実行後にその時点でのCPUのレジスタや特定の番地の情報をホストコンピュータ1側に送信する(7)。これに応じて、ホストコンピュータ1側の制御プログラム5は、受信したそれらのデータをホストコンピュータ1側の表示装置3に表示する(8)。次いで、繰り返し指定かどうかを判定し(9)、NOならば処理を終了し、YESならばステップ(5)に戻る。

【0072】これにより、ホストコンピュータ1から携帯型端末装置2に送られてきたこれらのコマンドに対す

る結果は、携帯型端末装置2からホストコンピュータ1へ送信される。そして、ホストコンピュータ1側ではその情報を用いてユーザに対して、携帯型端末装置2の現在の情報を知らせるための表示等を行う。従って、端末装置2の内部状態の表示を、携帯型端末装置2が装備する画面ではなく、通信回線の先にあるホストコンピュータ1に送信することによって、デバッグ対象のプログラムの画面出力を邪魔することなくデバッグ作業を行うことができるとともに、デバッガに対する入力も携帯型端末装置2のユーザインタフェースに影響されることなく行える。

【0073】このように、上記実施例では開発プログラムが実行される携帯型端末装置2の持つ種々の制限に影響されることなく、デバッグに必要とされるアドレスの指定をその時々状況によって変化しがちである実アドレスの数値そのものではなく、ソースプログラムを作成したときの名称を使用できるようにすることによって、開発効率を向上させる開発システムを構築することができる。

【0074】なお、上記実施例では、ホストコンピュータ1のCPU13が、例えばインテル社製の8086を利用しているので、機械語プログラムでのブレークポイントのコードとして「Int 3h(機械語でCCh)」の命令コードを使用している。

【0075】上記実施例では、機械語に翻訳されたプログラムのデバッグ方法について説明したが、本発明は後述するようにプログラム実行時にインタプリタと呼ばれる解釈プログラムを中間において、実行時にプログラムを解釈しながら動作するシステムにも制御プログラムに解釈機能を付加することにより適用することができる。

〔第2実施例〕以下、本発明に係る第4の発明と第2実施例との対応およびその作用について説明する。

【0076】第4の発明は、図1～図3に示した構成を備えた情報処理システムにおいて、記憶されたシンボル情報を参照して一方の情報処理装置(ホストコンピュータ1)のデバッグ対象のプログラムが入力情報(シンボル名等)を実行される記憶領域の実アドレスに変換して中間言語レベルの実行プログラムのデバッグを実行処理することにより、デバッグ対象の一方の情報処理装置における中間言語レベルの実行プログラムのデバッグ処理時に可読性の高いソースプログラムのシンボル名(図16を参照)でアドレス指定しながら効率よく実行プログラムのデバッグを行うものである。

【0077】図15はインタプリタ方式の実行モジュールの構成を説明する図である。

【0078】この図に示すように、インタプリタを介して実行されるプログラムの場合、実行速度の向上やプログラムサイズの縮小を目的としてソースプログラムを中間言語と呼ばれる形態のプログラムに変換して使用することが多い。中間言語に変換されたプログラムは、実行

コードの部分と変数表のブロックから構成され、変換時に名前表が作成される。なお、中間言語に変換されたプログラムは、携帯型端末装置2に置かれ、後述する名前表はホスト側のメモリに置かれる。

【0079】図16はインタプリタ方式の実行モジュールの名前表の一例を示すデータ構造図である。

【0080】この図に示されるように、中間言語プログラムの名前表は、実行プログラム内にある名前表領域の先頭を基準として、それに対するオフセットアドレス、変数の属性、変数の長さ、名前の綴りから構成されている。また、関数、手続等の名前もそれぞれの属性とともにこの表に格納される。変数表では、携帯型端末装置2にローディングするためにモジュールを変換するとき、実行プログラムのサイズを小さくすることを目的として、名前の綴りの部分を削除する形で変換される。

【0081】図17は本発明の第2実施例を示す情報処理装置のプログラムデバッグ方法を適用するコンピュータシステムのプログラム構成を説明するブロック図である。

【0082】図において、101はホストコンピュータで、後述するコントロールユニットを有し、プログラムに基づいて種々のデータ処理を行う。102は携帯型端末装置で、ホストコンピュータ100とインタフェースケーブルを介して接続される。103は表示装置で、入力装置104から入力されたデータを表示するとともに、システム状態をプログラムに応じて表示する。なお、入力装置104は、キーボード104aとポインティングデバイス104bから構成されている場合を示す。

【0083】105はホスト側の制御プログラム（携帯型端末装置102との通信を行うプログラム）で、OS112にその実行状態が管理されている。ホスト側の制御プログラム105は、入力装置104からのユーザの入力の読み取りと、表示装置103に対する出力と、携帯型端末側の制御プログラム109に対する通信動作を行う。

【0084】106は名前表で、図16に示す内容が記憶されている。113は前記携帯型端末装置102のOSで、携帯型端末側の制御プログラム109、中間言語に変換された実行プログラム110の実行を管理する。107、108は通信制御部で、通信経路（例えばRS232C、LAN、無線等）を介してホストコンピュータ101で開発した実行プログラムを携帯型端末装置102に転送を制御する。携帯型端末装置102側の制御プログラム109は、通信経路を通じてホスト側の制御プログラム105から入力されたコマンド、データに従って各種の動作を行い、また、デバッグの対象となる実行プログラムに対する監視の処理を行う。111はインタプリタで、中間言語を解釈実行する。

【0085】図18は本発明の一実施例を示す情報処理

装置のプログラムデバッグ方法における第2の対象プログラム指定処理手順の一例を示すフローチャートである。なお、(1)～(4)は各ステップを示す。

【0086】ユーザがプログラムを指定すると、ホスト側の制御プログラム105は通信を行うためのデータパケットを図5に示したプロトコルに従って作成し、携帯型端末装置102側の制御プログラム109に対して送信する。

【0087】制御プログラム109は、指定されたプログラムを実行可能状態にするように、図3に示した2次記憶装置としてのROM23から実メモリとしてのRAM22へ展開する等、操作を行う。

【0088】ここで、CPU19は指定されたプログラムを実行することができるかどうかを判定し(1)、ROM23に指定プログラムが存在しない場合または実行に必要なメモリ容量をRAM22に確保できない等の場合により即実行可能な状態に設定できなかったと判定した場合は(3)、ステップ(4)に進み、その旨をエラーを表すコードとともに携帯型端末装置102からホストコンピュータ101へ送信して、処理を終了する。

【0089】一方、ステップ(1)の判定でYESの場合は、指定されたプログラムが展開された領域の先頭アドレスをホストコンピュータ101側の制御プログラム105に対して送信する(2)。次いで、ホストコンピュータ101側の制御プログラム105は、以後の作業の高速化を目的として、シンボル情報のプログラムの先頭をベースアドレスに構成されているアドレス部分を実アドレスに変換するシンボル情報の再構成を名前表106を参照しながら行い(3)、処理を終了する。

【0090】図19は本発明の一実施例を示す情報処理装置のプログラムデバッグ方法における第2のブレークポイント設定処理手順の一例を示すフローチャートである。なお、(1)～(5)は各ステップを示す。

【0091】ホストコンピュータ101に対しては、サブルーチンの名称等のシンボルを使用して、このシンボルに相当するアドレスに実行が移ったならば、そこで実行を中断するようにユーザが指令する(1)。これにより、ホストコンピュータ101側の制御プログラム105は、入力装置104から入力されたデータの読み取り動作を行い、シンボルによるアドレス指定がなされているかどうかを判定し(2)、NOならばステップ(4)以降に進み、YESならば制御プログラム105は、読み取ったデータを入力データ中のアドレス指定に使用されているシンボルを名前表106を参照して実番地に変換する(3)。変換した後に、実アドレスによってプログラムのブレークポイント設定指示するように、プロトコルに従って作成したデータパケットを携帯型の端末装置102の制御プログラム109に対して送信して(4)、処理を終了する。

【0092】次いで、携帯型端末装置102の制御プロ

グラム109は、受信したデータに従ってインタプリタ111に対して実行停止箇所の指定を出す。その指令を受けたインタプリタ111は、該実行番地を実行しようとしたときに、プログラムの解釈を中断するように設定する(5)。なお、当該設定は、中間言語プログラム中に実行の中断を指令するコードを書き込む(同時に書き換えられる以前のコードの保存も行う)、あるいはインタプリタ中に実行をブレークする番地、条件等を保存しておく。ここで設定されたブレークポイントは、インタプリタがそれを認識した時点で中間言語プログラムの解釈・実行は中断され、携帯型端末装置102の制御をインタプリタ111から制御プログラム109に復帰するように働く。なお、ブレークポイントの確認は図10に示した第1のブレークポイント確認処理に準ずるの説明は省略する。

【0093】図20は本発明の一実施例を示す情報処理装置のプログラムデバッグ方法における変数の参照処理手順の一例を示すフローチャートである。なお、(1)～(9)は各ステップを示す。

【0094】ホストコンピュータ101側の制御プログラム105が入力装置104からユーザによりアドレスまたはアドレスを示すシンボルが入力されると(1)、シンボル表現によるアドレス指定がなされているかどうかを判断し(2)、名前表106を参照しながらシンボルを実アドレスに変換する(3)。変換作業の後、メモリ内容をホストコンピュータ101に対して送信するよう指示コマンドと、そのアドレスおよびその範囲を通信経路を通じて携帯型端末装置102に送信する(4)。

【0095】データパケットを π 受信した携帯型端末装置102側の制御プログラム109は、データパケット中のアドレス指定に従ってそのアドレスのデータを読み出し(5)、範囲中かどうかを判定し(6)、NOならば処理を終了し、YESならば現在のアドレスポインタのデータを読み出し(7)、アドレスポインタをインクリメントし(8)、読み出したデータを携帯型端末装置102からホストコンピュータ101へ送信して(9)、ステップ(6)へ戻る。

【0096】なお、送信する範囲は、指定された変数の大きさによって決定され、名前表106の属性部を参照することで変数の大きさを知ることができる。そして、制御プログラム105は、受信したデータホスト側の表示装置103に対して出力する。この時、制御プログラム105は、ユーザの選択によってバイト単位、ワード単位、文字列とみなす(表示不可能な文字コードは適当な文字に置換する)等のフォーマットで出力することが可能となる。

【0097】図21は本発明の一実施例を示す情報処理装置のプログラムデバッグ方法における変数の書換え処理手順の一例を示すフローチャートである。なお、

(1)～(12)は各ステップを示す。

【0098】ホストコンピュータ101側の制御プログラム105が入力装置104からユーザによりアドレスまたはアドレスを示すシンボルが入力されると(1)、シンボル表現によるアドレス指定がなされているかどうかを判断し(2)、名前表106を参照しながらシンボルを実アドレスに変換する(3)。変換作業の後、メモリ内容をホストコンピュータ101に対して送信するよう指示コマンドと、そのアドレスを通信経路を通じて携帯型端末装置102に送信する(4)。

【0099】次いで、制御プログラム109は指定アドレスのデータを携帯型端末装置102からホストコンピュータ101に送信し(5)、表示装置103に受信した指定データを表示する(6)。ここで、ユーザが入力装置104から書換えデータを入力すると(7)、現在アドレスポインタのデータを変更し(8)、アドレスポインタをインクリメントし(9)、変更終了が指示されているかどうかを判断し(10)、YESならば処理を終了し、NOならば変更データを携帯型端末装置102からホストコンピュータ101へ送信し(11)、変更データを受信したら、制御プログラム105が表示装置103に表示して(12)、ステップ(6)に戻る。以後、ユーザが変更の終了を示す入力を行うか、変数の大きさを越えて入力を行おうとするまで、ユーザの入力したデータで携帯型端末装置102の指定の変数の内容を書き換え、指定アドレスを変更するという動作を継続する。新たなデータをメモリへ転送した後に、そのデータをホスト側に送り返す。ホストコンピュータ101側の制御プログラム105は、変更後のデータを受信した後に、変更前のデータと変更後のデータを、ホスト側の表示装置103に出力する。なお、入力されたデータがデータとして認められないものであった場合、ホスト側の制御プログラム105がその旨をユーザに対して通知し、この時携帯型端末装置102の制御プログラム109に対して変更データの送信は行われない。

【0100】なお、プログラムの実行開始処理およびステップ実行処理は、図13および図14に示した処理に準ずるので説明は省略するが、プログラム実行時、携帯型端末装置102の制御は、一旦制御プログラム109からインタプリタ111に移され、中間言語プログラムの解釈実行が終了、あるいは中断した時点で再び制御プリンタ109に制御が戻るものとする。

【0101】また、上記各実施例ではプログラムの実行プラットフォームであるホストコンピュータ101と通信経路により接続された携帯型端末装置102はそれぞれ別種の情報処理装置としてシステム構成される場合について説明したが、上記ホストコンピュータ101と携帯型端末装置102とが同種の情報処理装置としてシステム構成される場合にも本発明を適用できることはいうまでもない。

【0102】さらに、上記各実施例では、ホストコンピ

ュータ101と通信経路、例えばRS232Cインタフェースケーブル等により接続された携帯型端末装置102とが1対1に通信する場合を例として説明したが、ホストコンピュータ101と携帯型端末装置102とがLANによって両者が接続されているシステムにおいても、例えば非同期通信ではモデムを使用して本実施例のコマンドパケットの送受信により、同期通信ではモデム間モデムで転送パケットの送受信を行うことによって同様に本発明を適用できる。

【0103】また、イーサネット、TCP/IPのプロトコルを利用するシステムにおいても、上述したコマンドパケットを送受信することにより本発明を適用することができる。

【0104】以下、図22を参照しながら図9等にし示したシンボルの実アドレスへの変換処理の詳細手順について説明する。

【0105】図22は本発明に係る情報処理装置のプログラムデバッグ方法におけるシンボルの実アドレスへの変換処理の詳細手順の一例を示すフローチャートである。なお、(1)～(6)は各ステップを示す。

【0106】まず、ホスト側制御プリンタ5は、通信制御部7を介して端末装置側から通信された指定コマンドが存在するかどうかを判定し(1)、NOならばエラー処理を実行して処理を終了する。

【0107】一方、ステップ(1)の判定で、指定されたコマンドが存在する場合には、アドレス指定がシンボル名(「ABC」)で行われているかどうかを判定し(2)、NOならばコマンド指定時の数値アドレス「・・・h」を使用し(3)、ステップ(6)以降に進む。

【0108】一方、ステップ(2)の判定でYESの場合には、指定されたシンボルがシンボル情報部6に存在するかどうかを判定して(4)、NOならばエラー処理を実行して終了し、YESの場合にはシンボルに対応するアドレス(図1の例では「8000h」)を取り出し(5)、指定コマンド、アドレスよりパケットを作成して(6)、処理を終了し、後続するホストから端末側へ指定アドレスを送信する処理以降する。

【0109】なお、各実施例において、第2の情報処理装置を携帯型端末装置で構成する場合について説明したが、通信インタフェースを備える電子機器であれば本発明を適用できる。

【0110】

【発明の効果】以上説明したように、本発明の第1の発明によれば、入力手段からの入力情報が前記シンボル名であると判別手段が判別した場合に、変換手段が前記シンボル情報記憶手段に記憶されたシンボル名を参照して実アドレス情報に変換するので、デバッグ対象の一方の情報処理装置における実行プログラムのデバッグ処理時に可読性の高いソースプログラムのシンボル名でアドレス指定を行うことができる。

【0111】第2の発明によれば、実行プログラムに記述されているシンボル名またはアドレス指定情報が一方または他方の情報処理装置から入力されたと判別した場合に、前記シンボル情報記憶手段に記憶されたシンボル名を参照して実アドレス情報に変換し、一方または他方の情報処理装置側から入力されるシンボル名に応じて一方の情報処理装置で実行される実行プログラムの実アドレスを生成ので、デバッグ対象の一方の情報処理装置における実行プログラムのデバッグ処理時に可読性の高いソースプログラムのシンボル名でアドレス指定しながら効率よく実行プログラムのデバッグを行うことができる。

【0112】第3の発明によれば、一方の情報処理装置のデバッグ対象のプログラムが入力情報を実行される記憶領域の実アドレスに変換して機械語のレベルの実行プログラムのデバッグを実行処理するので、デバッグ対象の一方の情報処理装置における実行プログラムのデバッグ処理時に可読性の高いソースプログラムのシンボル名でアドレス指定しながら効率よく可読性の低い実行プログラムのデバッグを行うことができる。

【0113】第4の発明によれば、記憶されたシンボル情報を参照して一方の情報処理装置のデバッグ対象のプログラムが入力情報を実行される記憶領域の実アドレスに変換して中間言語レベルの実行プログラムのデバッグを実行処理するので、デバッグ対象の一方の情報処理装置における中間言語レベルの実行プログラムのデバッグ処理時に可読性の高いソースプログラムのシンボル名でアドレス指定しながら効率よく実行プログラムのデバッグを行うことができる。

【0114】従って、ソースプログラムを作成した時の名称を使用したデバッグ処理を行えるようになり、開発効率を向上させる開発システム環境を提供できるという効果を奏する。

【図面の簡単な説明】

【図1】本発明の第1実施例を示す情報処理システムの構成を説明する外観斜視図である。

【図2】図1に示したコンピュータシステムの制御構成を説明するブロック図である。

【図3】図1に示したコンピュータシステムの要部詳細ブロック図である。

【図4】図2に示したシンボル情報部に記憶されたシンボル情報の一例を示す図である。

【図5】本発明に係る情報処理システムのプログラムデバッグ方式におけるコマンドパケット作成のためのプロトコルの一例を示す図である。

【図6】図1に示したホストコンピュータのコマンド処理手順の一例を示すフローチャートである。

【図7】本発明に係る情報処理システムのプログラムデバッグ方式における転送プログラムの指定入力を説明する図である。

21

【図8】本発明の一実施例を示す情報処理システムのプログラムデバッグ方法における第1の対象プログラム指定処理手順の一例を示すフローチャートである。

【図9】本発明の一実施例を示す情報処理システムのプログラムデバッグ方法における第1のブレークポイント設定処理手順の一例を示すフローチャートである。

【図10】本発明の一実施例を示す情報処理システムのプログラムデバッグ方法における第1のブレークポイント確認処理手順の一例を示すフローチャートである。

【図11】本発明の一実施例を示す情報処理システムのプログラムデバッグ方法におけるメモリ参照処理手順の一例を示すフローチャートである。

【図12】本発明の一実施例を示す情報処理システムのプログラムデバッグ方法におけるメモリ書換え処理手順の一例を示すフローチャートである。

【図13】本発明の一実施例を示す情報処理システムのプログラムデバッグ方法におけるプログラム実行処理手順の一例を示すフローチャートである。

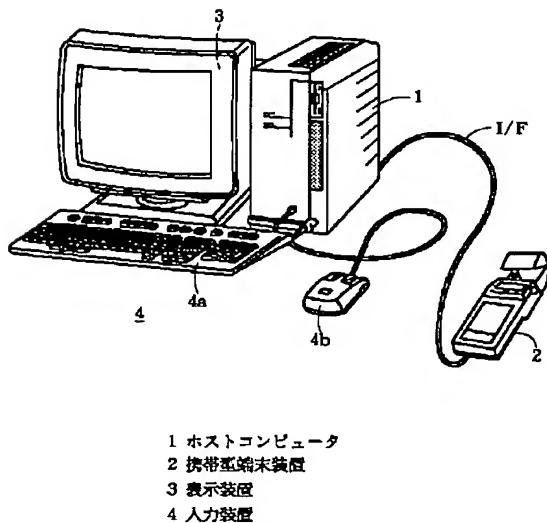
【図14】本発明の一実施例を示す情報処理システムのプログラムデバッグ方法におけるステップ実行処理手順の一例を示すフローチャートである。

【図15】インタプリタ方式の実行モジュールの構成を説明する図である。

【図16】インタプリタ方式の実行モジュールの名前表の一例を示すデータ構造図である。

【図17】本発明の第2実施例を示すプログラムデバッグ方法を適用するコンピュータシステムのプログラム構成を説明するブロック図である。

【図1】



22

【図18】本発明の一実施例を示す情報処理システムのプログラムデバッグ方法における第2の対象プログラム指定処理手順の一例を示すフローチャートである。

【図19】本発明の一実施例を示す情報処理システムのプログラムデバッグ方法における第2のブレークポイント設定処理手順の一例を示すフローチャートである。

【図20】本発明の一実施例を示す情報処理システムのプログラムデバッグ方法における変数の参照処理手順の一例を示すフローチャートである。

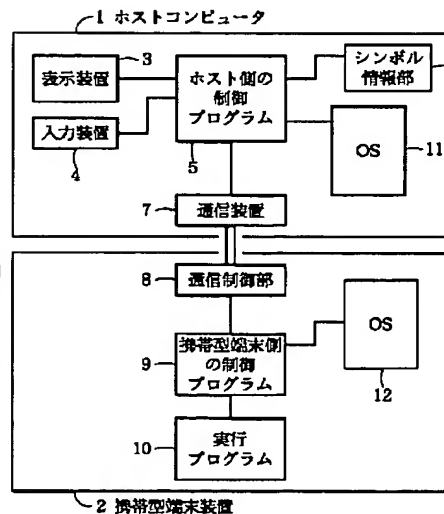
【図21】本発明の一実施例を示す情報処理システムのプログラムデバッグ方法における変数の書換え処理手順の一例を示すフローチャートである。

【図22】本発明に係る情報処理システムのプログラムデバッグ方法におけるシンボルの実アドレスへの変換処理の詳細手順の一例を示すフローチャートである。

【符号の説明】

- 1 ホストコンピュータ
- 2 携帯型端末装置
- 3 表示装置
- 4 入力装置
- 5 制御プログラム
- 6 シンボル情報部
- 7 通信制御部
- 8 通信制御部
- 9 制御プログラム
- 10 実行プログラム
- 11 OS
- 12 OS

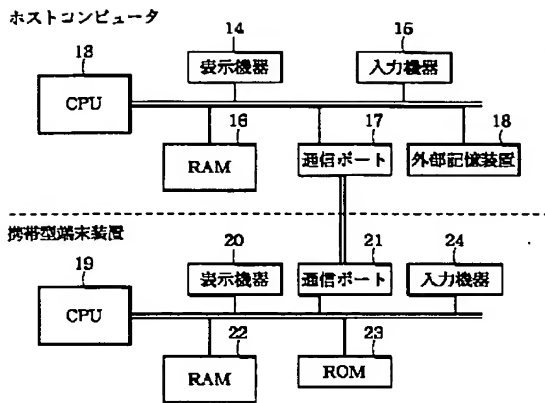
【図2】



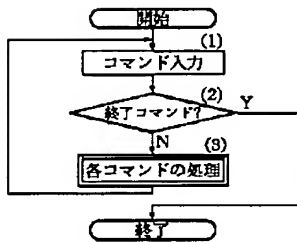
【図4】

アドレス	シンボル名
0000:0000	main000
0094:0084	PRT_END
0094:0080	PRT_ST
0094:0022	RAND_WK
0094:0026	STACKUP
0094:0024	TOX_PTR
0094:C536	_clib_seg
0094:B004	_count
0094:BD2E	_cur_grp
0094:BD06	_dic
0094:BD30	_dic_idx
0000:0652	_dump
0000:08BD	_egcfind
0000:0885	_egcfiniah
0000:0880	_egcinit
0000:08C1	_egcnext
0000:08C5	_egcread
0000:08B5	_egcdadd
0000:08B9	_egcdaddl
0094:0048	_errmsg

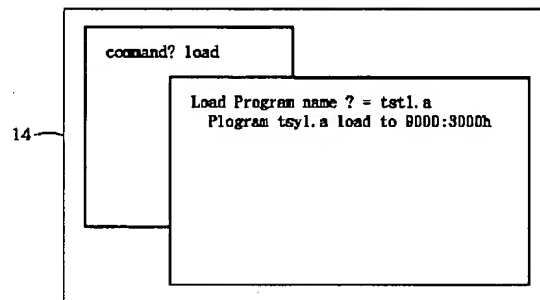
【図3】



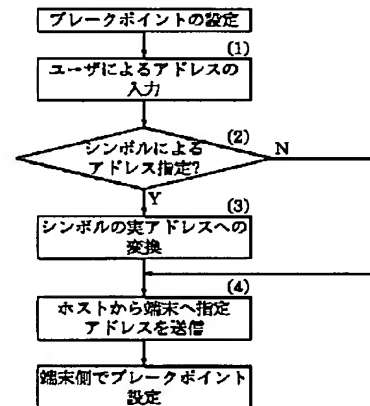
【図6】



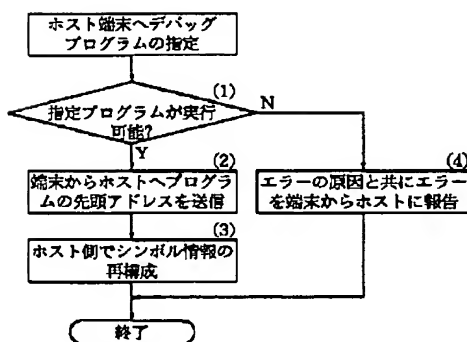
【図7】



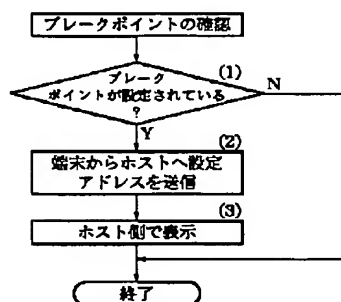
【図9】



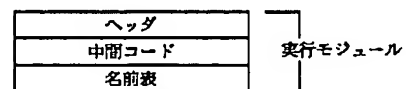
【図8】



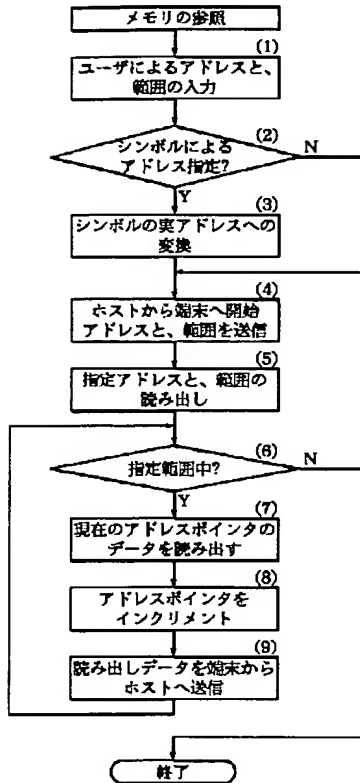
【図10】



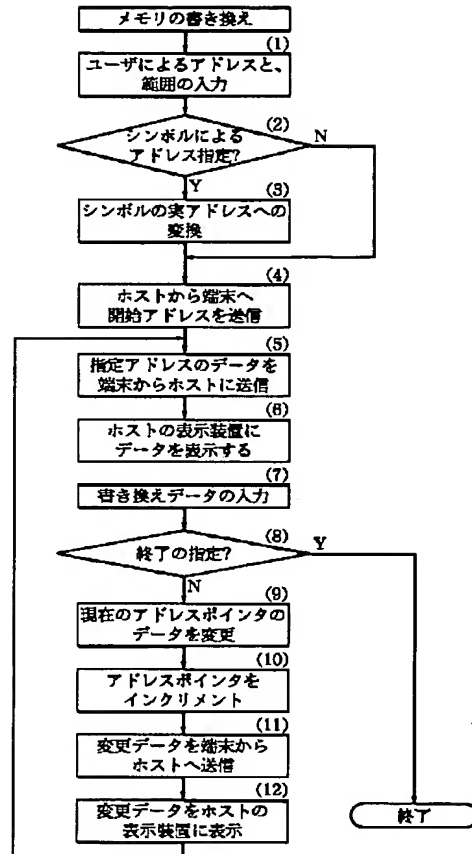
【図15】



【図11】



【図12】

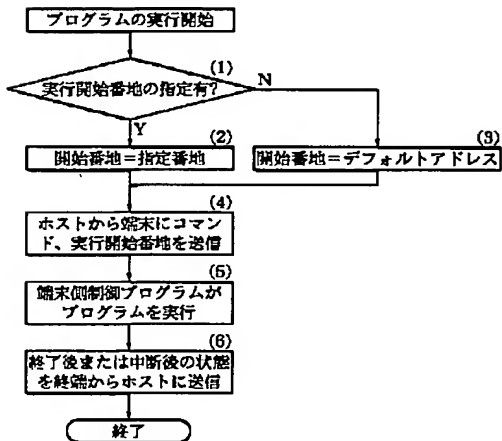


【図16】

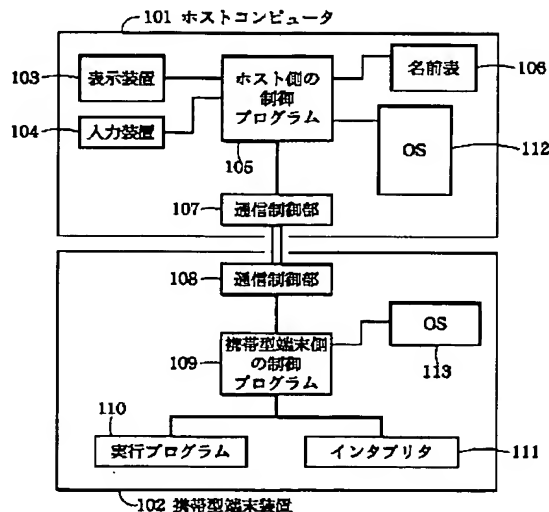
	size	address
Program	02EB	
text block	0D57	0010
table block of	0154	0D67
region of variables	0099A	

Elements	Length	Address	Name
2	0000:4	ACT	
200	0070:2	BLFS	
2	0005:6	DMO	
2	0005:C	FS	
2	0000:2	PLNC	
2	0003:0	HAN3	
2	0004:0	HAN4	
2	0003:4	HKAKU3	
2	0004:4	HKAKU4	
2	0005:E	HNDRET	
2	0001:2	KX	
2	0001:E	KX2	
2	0002:C	KX3	
2	0003:C	KX6	
2	0004:C	KX7	
2	0001:4	KY	
2	0002:0	KY2	
2	0002:E	KY3	
2	0003:E	KY6	
2	0004:E	KY7	
8	0095:A	LPS	
8	0097:2	MAIF	
2	0000:6	MO	
2	0005:0	MTP	
2	0002:A	NPT	
2	0003:A	NPT3	
2	0004:A	NPT4	
2	0000:8	PSIZE	
2	0001:C	SH	
2	0002:8	SE2	
2	0003:6	SE3	
2	0004:6	SE4	
2	0003:8	SFMS	
2	0004:8	SFMS	
2	0003:2	SKAKU3	
2	0004:2	SKAKU4	
2	0001:A	SS	
2	0002:6	SS2	
2	0000:0	STAT	

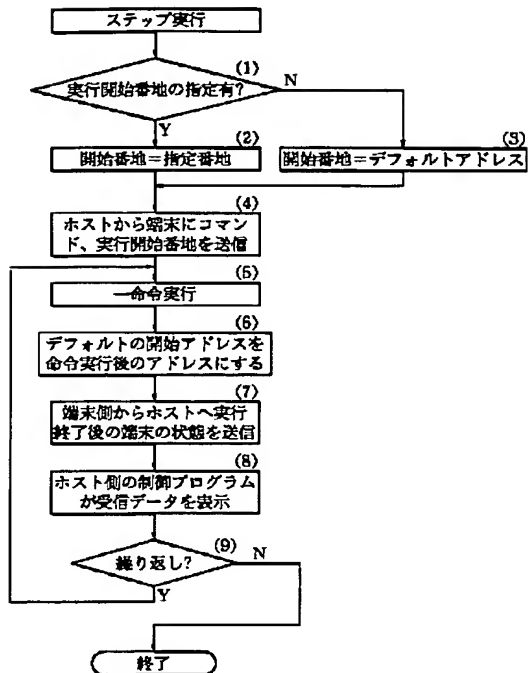
【図13】



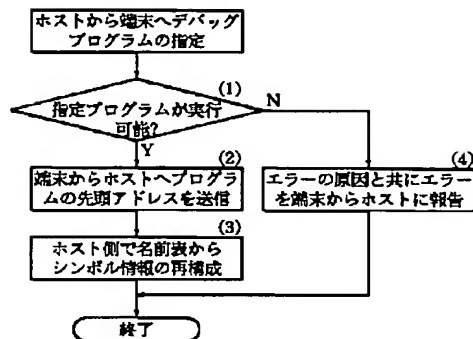
【図17】



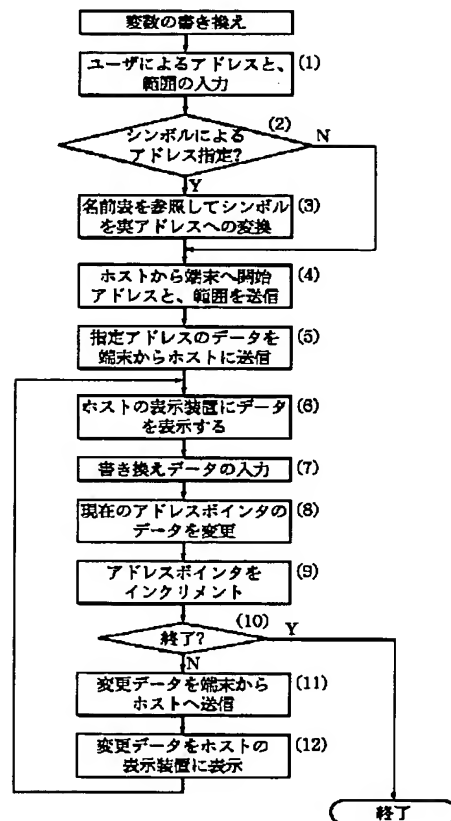
【図 14】



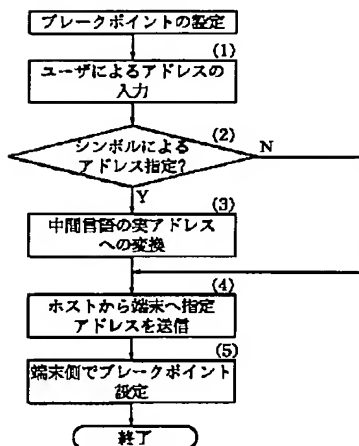
【図 18】



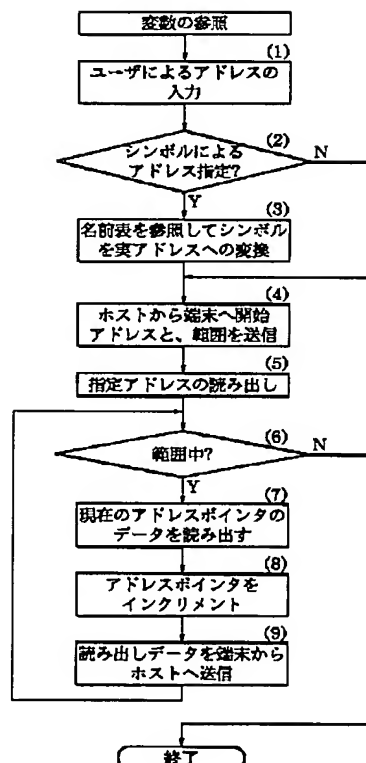
【図 21】



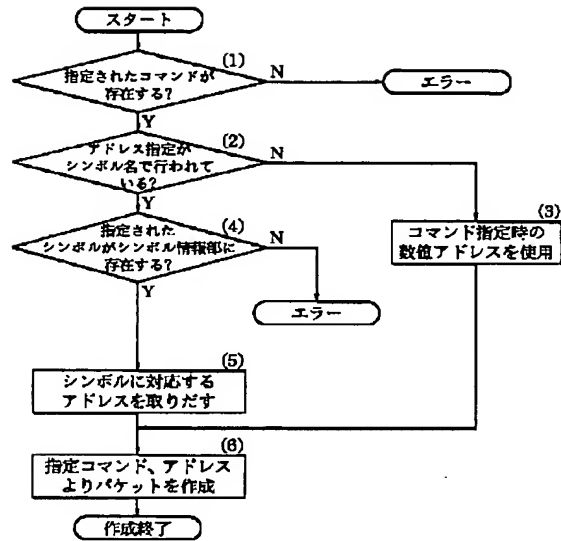
【図 19】



【図 20】



【図22】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ BLACK BORDERS
- ☒ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☒ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.